

```
In [1]: # Anova diseño multifactorial 2^k
```

```
# 1. Carga inicial de datos.
```

```
if(!require(psych)){install.packages("psych")}
if(!require(FSA)){install.packages("FSA")}
if(!require(ggplot2)){install.packages("ggplot2")}
if(!require(car)){install.packages("car")}
if(!require(multcompView)){install.packages("multcompView")}
if(!require(lsmmeans)){install.packages("lsmmeans")}
if(!require(rcompanion)){install.packages("rcompanion")}
```

```
ln <- ("
```

Algoritmo	Entrenamiento	Rendimiento	Acelerador
'Algoritmo A'	MT500	12000	NA-NA
'Algoritmo A'	MT500	14005	NA-NA
'Algoritmo A'	MT500	13508	NA-NA
'Algoritmo A'	MT500	9503	NA-NA
'Algoritmo A'	MT500	14004	NA-NA
'Algoritmo A'	MT1000	11502	NA-NA
'Algoritmo A'	MT1000	13006	NA-NA
'Algoritmo A'	MT1000	13252	NA-NA
'Algoritmo A'	MT1000	14253	NA-NA
'Algoritmo A'	MT1000	15003	NA-NA
'Algoritmo A'	MT5000	12504	NA-NA
'Algoritmo A'	MT5000	11504	NA-NA
'Algoritmo A'	MT5000	9500	NA-NA
'Algoritmo A'	MT5000	11506	NA-NA
'Algoritmo A'	MT5000	16000	NA-NA
'Algoritmo A'	MT50000	13008	NA-NA
'Algoritmo A'	MT50000	10506	NA-NA
'Algoritmo A'	MT50000	13005	NA-NA
'Algoritmo A'	MT50000	17002	NA-NA
'Algoritmo A'	MT50000	13008	NA-NA
'Algoritmo B'	MT500	11005	NA-NA
'Algoritmo B'	MT500	12007	NA-NA
'Algoritmo B'	MT500	12509	NA-NA
'Algoritmo B'	MT500	10504	NA-NA
'Algoritmo B'	MT500	12002	NA-NA
'Algoritmo B'	MT1000	12504	NA-NA
'Algoritmo B'	MT1000	13501	NA-NA
'Algoritmo B'	MT1000	13501	NA-NA
'Algoritmo B'	MT1000	13252	NA-NA
'Algoritmo B'	MT1000	15256	NA-NA
'Algoritmo B'	MT5000	12253	NA-NA
'Algoritmo B'	MT5000	11255	NA-NA
'Algoritmo B'	MT5000	10006	NA-NA
'Algoritmo B'	MT5000	11252	NA-NA
'Algoritmo B'	MT5000	14004	NA-NA
'Algoritmo B'	MT50000	12007	NA-NA
'Algoritmo B'	MT50000	11505	NA-NA
'Algoritmo B'	MT50000	14009	NA-NA
'Algoritmo B'	MT50000	15000	NA-NA
'Algoritmo B'	MT50000	12009	NA-NA
'Algoritmo C'	MT500	9000	NA-NA
'Algoritmo C'	MT500	11003	NA-NA
'Algoritmo C'	MT500	11505	NA-NA
'Algoritmo C'	MT500	9509	NA-NA
'Algoritmo C'	MT500	11003	NA-NA
'Algoritmo C'	MT1000	11508	NA-NA
'Algoritmo C'	MT1000	12508	NA-NA
'Algoritmo C'	MT1000	12506	NA-NA
'Algoritmo C'	MT1000	12254	NA-NA
'Algoritmo C'	MT1000	13253	NA-NA
'Algoritmo C'	MT5000	11255	NA-NA
'Algoritmo C'	MT5000	10257	NA-NA
'Algoritmo C'	MT5000	9500	NA-NA
'Algoritmo C'	MT5000	9255	NA-NA
'Algoritmo C'	MT5000	12009	NA-NA
'Algoritmo C'	MT50000	11000	NA-NA
'Algoritmo C'	MT50000	9509	NA-NA
'Algoritmo C'	MT50000	13009	NA-NA
'Algoritmo C'	MT50000	14005	NA-NA
'Algoritmo C'	MT50000	11001	NA-NA

'Algoritmo A'	MT500	12046	NA-SW
'Algoritmo A'	MT500	14589	NA-SW
'Algoritmo A'	MT500	13723	NA-SW
'Algoritmo A'	MT500	9799	NA-SW
'Algoritmo A'	MT500	14715	NA-SW
'Algoritmo A'	MT1000	11144	NA-SW
'Algoritmo A'	MT1000	13920	NA-SW
'Algoritmo A'	MT1000	13226	NA-SW
'Algoritmo A'	MT1000	14845	NA-SW
'Algoritmo A'	MT1000	15142	NA-SW
'Algoritmo A'	MT5000	12352	NA-SW
'Algoritmo A'	MT5000	11296	NA-SW
'Algoritmo A'	MT5000	9737	NA-SW
'Algoritmo A'	MT5000	11129	NA-SW
'Algoritmo A'	MT5000	16409	NA-SW
'Algoritmo A'	MT50000	13872	NA-SW
'Algoritmo A'	MT50000	10100	NA-SW
'Algoritmo A'	MT50000	13419	NA-SW
'Algoritmo A'	MT50000	17398	NA-SW
'Algoritmo A'	MT50000	13164	NA-SW
'Algoritmo B'	MT500	11047	NA-SW
'Algoritmo B'	MT500	12226	NA-SW
'Algoritmo B'	MT500	12105	NA-SW
'Algoritmo B'	MT500	10418	NA-SW
'Algoritmo B'	MT500	12446	NA-SW
'Algoritmo B'	MT1000	12156	NA-SW
'Algoritmo B'	MT1000	13968	NA-SW
'Algoritmo B'	MT1000	13891	NA-SW
'Algoritmo B'	MT1000	13778	NA-SW
'Algoritmo B'	MT1000	15448	NA-SW
'Algoritmo B'	MT5000	12441	NA-SW
'Algoritmo B'	MT5000	11767	NA-SW
'Algoritmo B'	MT5000	10340	NA-SW
'Algoritmo B'	MT5000	11306	NA-SW
'Algoritmo B'	MT5000	14565	NA-SW
'Algoritmo B'	MT50000	12725	NA-SW
'Algoritmo B'	MT50000	11169	NA-SW
'Algoritmo B'	MT50000	14749	NA-SW
'Algoritmo B'	MT50000	15566	NA-SW
'Algoritmo B'	MT50000	12239	NA-SW
'Algoritmo C'	MT500	9082	NA-SW
'Algoritmo C'	MT500	11887	NA-SW
'Algoritmo C'	MT500	11799	NA-SW
'Algoritmo C'	MT500	9300	NA-SW
'Algoritmo C'	MT500	11049	NA-SW
'Algoritmo C'	MT1000	11378	NA-SW
'Algoritmo C'	MT1000	12659	NA-SW
'Algoritmo C'	MT1000	12905	NA-SW
'Algoritmo C'	MT1000	12782	NA-SW
'Algoritmo C'	MT1000	13196	NA-SW
'Algoritmo C'	MT5000	11795	NA-SW
'Algoritmo C'	MT5000	10316	NA-SW
'Algoritmo C'	MT5000	9947	NA-SW
'Algoritmo C'	MT5000	9420	NA-SW
'Algoritmo C'	MT5000	12699	NA-SW
'Algoritmo C'	MT50000	11024	NA-SW
'Algoritmo C'	MT50000	9556	NA-SW
'Algoritmo C'	MT50000	13900	NA-SW
'Algoritmo C'	MT50000	14006	NA-SW
'Algoritmo C'	MT50000	11738	NA-SW
'Algoritmo A'	MT500	126572	HW-NA
'Algoritmo A'	MT500	140058	HW-NA
'Algoritmo A'	MT500	139580	HW-NA
'Algoritmo A'	MT500	92583	HW-NA
'Algoritmo A'	MT500	148057	HW-NA
'Algoritmo A'	MT1000	110078	HW-NA
'Algoritmo A'	MT1000	131942	HW-NA
'Algoritmo A'	MT1000	133797	HW-NA
'Algoritmo A'	MT1000	140026	HW-NA
'Algoritmo A'	MT1000	155479	HW-NA
'Algoritmo A'	MT5000	125809	HW-NA
'Algoritmo A'	MT5000	114264	HW-NA
'Algoritmo A'	MT5000	98797	HW-NA
'Algoritmo A'	MT5000	113400	HW-NA
'Algoritmo A'	MT5000	168898	HW-NA
'Algoritmo A'	MT50000	133452	HW-NA

'Algoritmo A'	MT50000	101641	HW-NA
'Algoritmo A'	MT50000	133155	HW-NA
'Algoritmo A'	MT50000	175156	HW-NA
'Algoritmo A'	MT50000	131945	HW-NA
'Algoritmo B'	MT500	110317	HW-NA
'Algoritmo B'	MT500	129244	HW-NA
'Algoritmo B'	MT500	127966	HW-NA
'Algoritmo B'	MT500	109783	HW-NA
'Algoritmo B'	MT500	122936	HW-NA
'Algoritmo B'	MT1000	128830	HW-NA
'Algoritmo B'	MT1000	134437	HW-NA
'Algoritmo B'	MT1000	138321	HW-NA
'Algoritmo B'	MT1000	132000	HW-NA
'Algoritmo B'	MT1000	157693	HW-NA
'Algoritmo B'	MT5000	121964	HW-NA
'Algoritmo B'	MT5000	119872	HW-NA
'Algoritmo B'	MT5000	106654	HW-NA
'Algoritmo B'	MT5000	112666	HW-NA
'Algoritmo B'	MT5000	145535	HW-NA
'Algoritmo B'	MT50000	127938	HW-NA
'Algoritmo B'	MT50000	115179	HW-NA
'Algoritmo B'	MT50000	143021	HW-NA
'Algoritmo B'	MT50000	150357	HW-NA
'Algoritmo B'	MT50000	121216	HW-NA
'Algoritmo C'	MT500	95474	HW-NA
'Algoritmo C'	MT500	113776	HW-NA
'Algoritmo C'	MT500	117473	HW-NA
'Algoritmo C'	MT500	92900	HW-NA
'Algoritmo C'	MT500	115582	HW-NA
'Algoritmo C'	MT1000	115279	HW-NA
'Algoritmo C'	MT1000	122184	HW-NA
'Algoritmo C'	MT1000	124770	HW-NA
'Algoritmo C'	MT1000	128403	HW-NA
'Algoritmo C'	MT1000	135219	HW-NA
'Algoritmo C'	MT5000	112562	HW-NA
'Algoritmo C'	MT5000	108736	HW-NA
'Algoritmo C'	MT5000	91064	HW-NA
'Algoritmo C'	MT5000	98171	HW-NA
'Algoritmo C'	MT5000	120277	HW-NA
'Algoritmo C'	MT50000	111299	HW-NA
'Algoritmo C'	MT50000	90193	HW-NA
'Algoritmo C'	MT50000	135178	HW-NA
'Algoritmo C'	MT50000	146158	HW-NA
'Algoritmo C'	MT50000	113845	HW-NA
'Algoritmo A'	MT500	124252	HW-SW
'Algoritmo A'	MT500	143833	HW-SW
'Algoritmo A'	MT500	138907	HW-SW
'Algoritmo A'	MT500	91010	HW-SW
'Algoritmo A'	MT500	143901	HW-SW
'Algoritmo A'	MT1000	116563	HW-SW
'Algoritmo A'	MT1000	136455	HW-SW
'Algoritmo A'	MT1000	130411	HW-SW
'Algoritmo A'	MT1000	140060	HW-SW
'Algoritmo A'	MT1000	154308	HW-SW
'Algoritmo A'	MT5000	124480	HW-SW
'Algoritmo A'	MT5000	111552	HW-SW
'Algoritmo A'	MT5000	99135	HW-SW
'Algoritmo A'	MT5000	110208	HW-SW
'Algoritmo A'	MT5000	167228	HW-SW
'Algoritmo A'	MT50000	134267	HW-SW
'Algoritmo A'	MT50000	102119	HW-SW
'Algoritmo A'	MT50000	138036	HW-SW
'Algoritmo A'	MT50000	171632	HW-SW
'Algoritmo A'	MT50000	130666	HW-SW
'Algoritmo B'	MT500	116942	HW-SW
'Algoritmo B'	MT500	129721	HW-SW
'Algoritmo B'	MT500	128834	HW-SW
'Algoritmo B'	MT500	100390	HW-SW
'Algoritmo B'	MT500	127771	HW-SW
'Algoritmo B'	MT1000	121789	HW-SW
'Algoritmo B'	MT1000	135311	HW-SW
'Algoritmo B'	MT1000	136587	HW-SW
'Algoritmo B'	MT1000	139664	HW-SW
'Algoritmo B'	MT1000	151543	HW-SW
'Algoritmo B'	MT5000	128962	HW-SW
'Algoritmo B'	MT5000	110157	HW-SW

'Algoritmo B'	MT5000	106129	HW-SW
'Algoritmo B'	MT5000	114634	HW-SW
'Algoritmo B'	MT5000	143337	HW-SW
'Algoritmo B'	MT50000	129292	HW-SW
'Algoritmo B'	MT50000	117502	HW-SW
'Algoritmo B'	MT50000	143687	HW-SW
'Algoritmo B'	MT50000	153488	HW-SW
'Algoritmo B'	MT50000	129773	HW-SW
'Algoritmo C'	MT500	99920	HW-SW
'Algoritmo C'	MT500	110833	HW-SW
'Algoritmo C'	MT500	117879	HW-SW
'Algoritmo C'	MT500	96441	HW-SW
'Algoritmo C'	MT500	119688	HW-SW
'Algoritmo C'	MT1000	117995	HW-SW
'Algoritmo C'	MT1000	122984	HW-SW
'Algoritmo C'	MT1000	120317	HW-SW
'Algoritmo C'	MT1000	120213	HW-SW
'Algoritmo C'	MT1000	137806	HW-SW
'Algoritmo C'	MT5000	117014	HW-SW
'Algoritmo C'	MT5000	105529	HW-SW
'Algoritmo C'	MT5000	98755	HW-SW
'Algoritmo C'	MT5000	96010	HW-SW
'Algoritmo C'	MT5000	126548	HW-SW
'Algoritmo C'	MT50000	113527	HW-SW
'Algoritmo C'	MT50000	99385	HW-SW
'Algoritmo C'	MT50000	136573	HW-SW
'Algoritmo C'	MT50000	141965	HW-SW
'Algoritmo C'	MT50000	111994	HW-SW

)

Se introduce la tabla.

```
Data <- read.table(textConnection(ln), header=TRUE)
```

Se ordenan los datos según los ingresamos. (Evitar orden alfabético por R).

```
Data$Entrenamiento <- factor(Data$Entrenamiento, levels = unique(Data$Entrenamiento))
```

```
Data$Acelerador <- factor(Data$Acelerador, levels = unique(Data$Acelerador))
```

```
Data$Algoritmo <- factor(Data$Algoritmo, levels = unique(Data$Algoritmo))
```

2. Verificación de la lectura de datos.

```
library(psych)
```

```
headTail(Data)
```

```
str(Data)
```

```
summary(Data)
```

```
rm(ln)
```

3. Gráfico simple de interacción.

Variable dependiente: Rendimiento

Variables independientes: Algoritmo y Método de Entrenamiento.

```
interaction.plot(x.factor = Data$Entrenamiento,
```

```
trace.factor = Data$Algoritmo,
```

```
response = Data$Rendimiento,
```

```
fun = mean,
```

```
type = "b",
```

```
col = c("black", "red", "green"),
```

```
pch = c(19,17,15),
```

```
fixed = TRUE,
```

```
leg.bty = "o")
```

4. Se realiza cambio al gráfico para agregar el acelerador.

Algoritmo en función del acelerador para ver el rendimiento.

```
interaction.plot(x.factor = Data$Acelerador,
```

```
trace.factor = Data$Algoritmo,
```

```
response = Data$Rendimiento,
```

```
fun = mean,
```

```
type = "b",
```

```
col = c("black", "red", "green"),
```

```
pch = c(19, 17, 15),
```

```
fixed = TRUE,
```

```
leg.bty = "o")
```

5. Modelo Lineal y anova

```

# * Analisis de factores e interacciones de los factores.
model <- lm(Rendimiento ~ Entrenamiento * Algoritmo * Acelerador, data = Data)
library(car)
Anova(model, type = "II")

# Hay diferencias entre los grupos de entrenamiento, los algoritmos y el acelerador.
# Las factores impactan la variable de respuesta.
# Las interacciones no impactan las variables de respuesta | Algoritmo:Acelerador

# 6. Evaluación de los supuestos

x <- residuals(model)
library(rcompanion)
plotNormalHistogram(x)
plot(fitted(model), residuals(model))
plot(model)

# Parece haber normalidad, no hay homocedasticidad.

# Se procede a hacer transformación iniciando desde la forma menos agresiva a la más agresiva, hasta cumplir

# 7. Transformación por raíz cuadrada

library(rcompanion)
T_sqrt <- sqrt(Data$Rendimiento) # Ingresar variable dependiente.
model <- lm(T_sqrt ~ Entrenamiento * Algoritmo * Acelerador, data = Data)

library(car)
Anova(model, type = "II")

# Supuestos

x <- residuals(model)
library(rcompanion)
plotNormalHistogram(x)
plot(fitted(model), residuals(model))
plot(model)

# Estamos llegando a conclusiones no al rendimiento, sino a la raíz cuadrada del rendimiento, porque eso es

# 8. Transformación por raíz cúbica

library(rcompanion)
T_cub <- sign(Data$Rendimiento) * abs(Data$Rendimiento)^(1/3) # Ingresar variable dependiente.
model <- lm(T_cub ~ Entrenamiento * Algoritmo * Acelerador, data = Data)

library(car)
Anova(model, type = "II")

# Supuestos

x <- residuals(model)
library(rcompanion)
plotNormalHistogram(x)
plot(fitted(model), residuals(model))
plot(model)

# 9. Transformación por Logaritmo

library(rcompanion)
T_log <- log(Data$Rendimiento)

model <- lm(T_log ~ Entrenamiento * Algoritmo * Acelerador, data = Data)
library(car)
Anova(model, type = "II")

x <- residuals(model)
library(rcompanion)

plotNormalHistogram(x)
plot(fitted(model), residuals(model))
plot(model)

# 10. Prueba Levene

```

```

leveneTest(T_log ~ Entrenamiento * Algoritmo * Acelerador, data = Data)

# 11. Analisis post-hoc por algoritmo

library(lsmeans)
marginal <- lsmeans(model, pairwise ~ Algoritmo, adjust="tukey")

library(multcomp)
CLD <- cld(marginal, alpha = 0.05, Letters = letters, adjust = "tukey")
CLD

# 12. Analisis post-hoc por entrenamiento

library(lsmeans)
marginal <- lsmeans(model, pairwise ~ Entrenamiento, adjust="tukey")

library(multcomp)
CLD <- cld(marginal, alpha = 0.05, Letters = letters, adjunst = "tukey")
CLD

# 13. Análisis post-hoc para acelerador 2^k

library(lsmeans)
marginal <- lsmeans(model, pairwise ~ Acelerador, adjust="tukey")

library(multcomp)
CLD <- cld(marginal, alpha = 0.05, Letters = letters, adjunst = "tukey")
CLD

# 14. Gráficos finales.

library(FSA)
Sum <- Summarize(T_log ~ Entrenamiento + Algoritmo, data = Data, digits = 3)

# Se agrega el se
Sum$se <- Sum$sd / sqrt(Sum$n)
Sum$se <- signif(Sum$se, digits = 3)
Sum

### Ordenamos

Sum$Entrenamiento <- factor(Sum$Entrenamiento, levels = unique(Sum$Entrenamiento))

### Graficamos

library(ggplot2)
pd <- position_dodge(.2)

ggplot(Sum, aes(x = Entrenamiento,
                y = mean, color = Algoritmo)) +
  geom_errorbar(aes(ymin = mean - se, ymax = mean + se), width = .2, size = 0.7, position = pd) +
  geom_point(shape = 15, size = 4, position = pd) +
  theme_bw() +
  theme(axis.title = element_text(face = "bold")) +
  scale_colour_manual(values = c("black", "red", "green")) +
  ylab("Logaritmo de rendimiento")

# Para acelerador

Sum <- Summarize(T_log ~ Acelerador + Algoritmo, data = Data, digits = 3)

# Se agrega el se
Sum$se <- Sum$sd / sqrt(Sum$n)
Sum$se <- signif(Sum$se, digits = 3)
Sum

### Ordenamos | Correccion: se cambia entrenamiento por acelerador
Sum$Acelerador <- factor(Sum$Acelerador, levels = unique(Sum$Acelerador))

### Graficamos

ggplot(Sum, aes(x = Acelerador,
                y = mean, color = Algoritmo)) +
  geom_errorbar(aes(ymin = mean - se, ymax = mean + se), width = .2, size = 0.7, position = pd) +

```

```

geom_point(shape = 15, size = 4, position = pd) +
theme_bw() +
theme(axis.title = element_text(face = "bold")) +
scale_colour_manual(values = c("black", "red", "green")) +
ylab("Logaritmo de rendimiento")

# Bigotes pequeños por ser dato transformados

# 15. Gráfico de promedios transformados

### Creamos un dato llamado sum con promedios y se
library(FSA)
Sum <- Summarize(T_log ~ Algoritmo, data = Data, digits = 3)

### Agregamos el se
Sum$se <- Sum$sd / sqrt(Sum$n)
Sum$se <- signif(Sum$se, digits = 3)
Sum

### Ordenamos / Correccion: se cambia entrenamiento por algoritmo
Sum$Algoritmo <- factor(Sum$Algoritmo, levels = unique(Sum$Algoritmo))

### Graficamos
library(ggplot2)
pd <- position_dodge(.2)

# Correccion: se cambia entrenamiento por algoritmo
ggplot(Sum, aes(x = Algoritmo,
                y = mean, color = Algoritmo)) +
  geom_errorbar(aes(ymin = mean - se, ymax = mean + se), width = .2, size = 0.7, position = pd) +
  geom_point(shape = 15, size = 4, position = pd) +
  theme_bw() +
  theme(axis.title = element_text(face = "bold")) +
  scale_colour_manual(values = c("black", "red", "green")) +
  ylab("Logaritmo de rendimiento")

# 16. Des-transformando promedios

library(FSA)
# Corrección, no se agrega Entrenamiento. El grafico final no seria el mismo en la presentacion.
Sum <- Summarize(T_log ~ Algoritmo, data = Data, digits = 3)

Sum$mean <- exp(Sum$mean)
Sum$sd <- exp(Sum$sd)

### Agregamos el se
Sum$se <- Sum$sd / sqrt(Sum$n)
Sum$se <- signif(Sum$se, digits = 3)
Sum

ggplot(Sum, aes(x = Algoritmo,
                y = mean, color = Algoritmo)) +
  geom_errorbar(aes(ymin = mean - se, ymax = mean + se), width = .2, size = 0.7, position = pd) +
  geom_point(shape = 15, size = 4, position = pd) +
  theme_bw() +
  theme(axis.title = element_text(face = "bold")) +
  scale_colour_manual(values = c("black", "red", "green")) +
  ylab("Logaritmo de rendimiento")

```

Loading required package: psych

Loading required package: FSA

FSA v0.9.4. See citation('FSA') if used in publication.
Run fishR() for related website and fishR('IFAR') for related book.

Attaching package: 'FSA'

The following object is masked from 'package:psych':

headtail

Loading required package: ggplot2

Attaching package: 'ggplot2'

The following objects are masked from 'package:psych':

%+%, alpha

Loading required package: car

Loading required package: carData

Registered S3 methods overwritten by 'car':

method	from
hist.boot	FSA
confint.boot	FSA

Attaching package: 'car'

The following object is masked from 'package:FSA':

bootCase

The following object is masked from 'package:psych':

logit

Loading required package: multcompView

Loading required package: lsmeans

Loading required package: emmeans

The 'lsmeans' package is now basically a front end for 'emmeans'.
Users are encouraged to switch the rest of the way.
See help('transition') for more information, including how to
convert old 'lsmeans' objects and scripts to work with 'emmeans'.

Loading required package: rcompanion

Attaching package: 'rcompanion'

The following object is masked from 'package:psych':

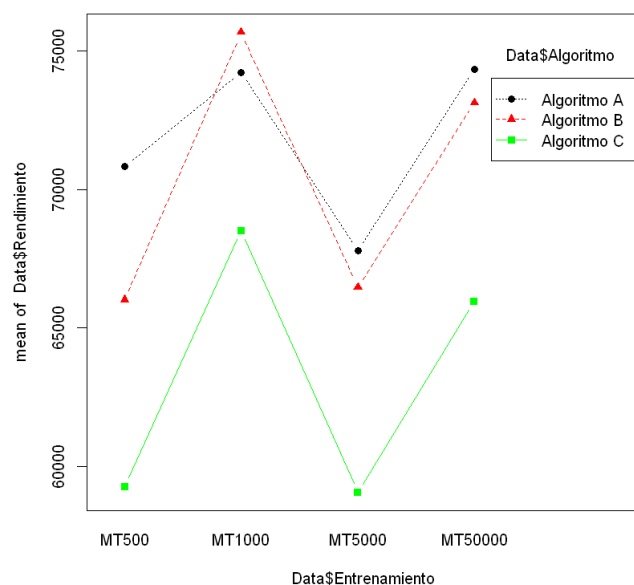
phi

A data.frame: 9 × 4

	Algoritmo	Entrenamiento	Rendimiento	Acelerador
	<fct>	<fct>	<chr>	<fct>
1	Algoritmo A	MT500	12000	NA-NA
2	Algoritmo A	MT500	14005	NA-NA
3	Algoritmo A	MT500	13508	NA-NA
4	Algoritmo A	MT500	9503	NA-NA
...	NA	NA	...	NA
237	Algoritmo C	MT50000	99385	HW-SW
238	Algoritmo C	MT50000	136573	HW-SW
239	Algoritmo C	MT50000	141965	HW-SW
240	Algoritmo C	MT50000	111994	HW-SW

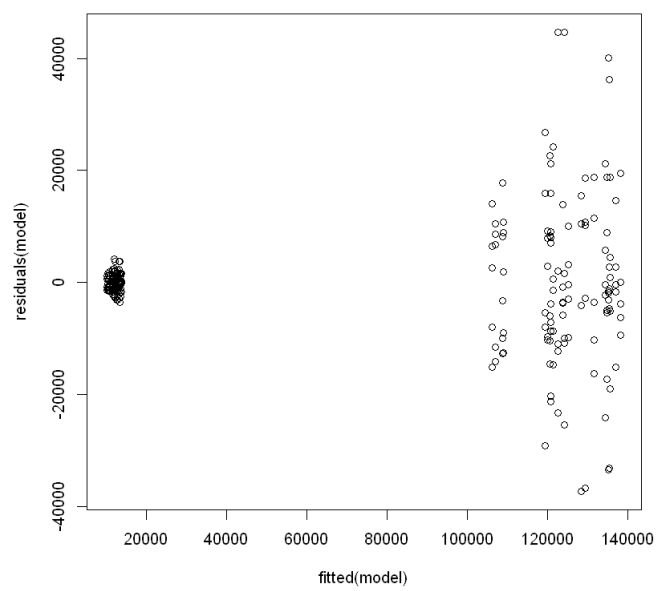
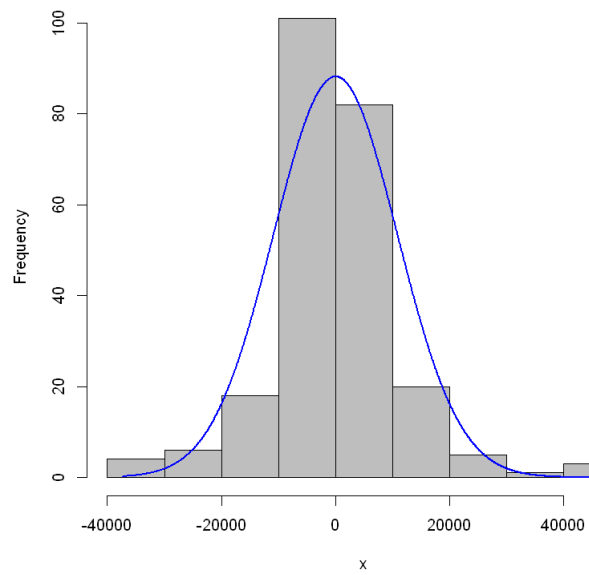
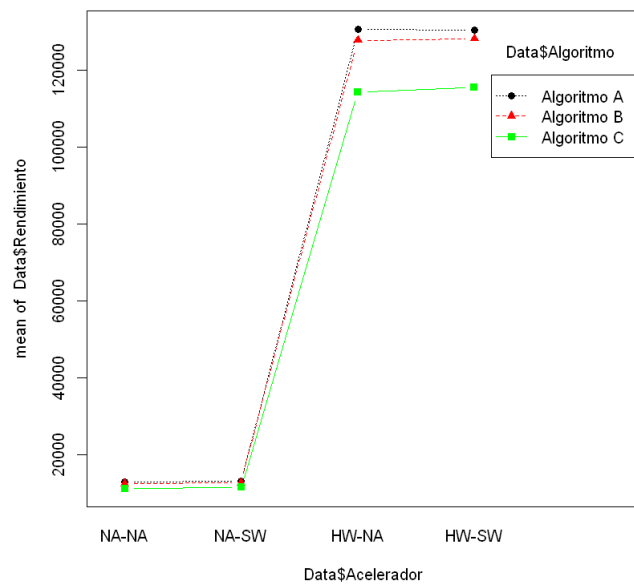
```
'data.frame':  240 obs. of  4 variables:
 $ Algoritmo      : Factor w/  3 levels "Algoritmo A",...: 1 1 1 1 1 1 1 1 1 1 ...
 $ Entrenamiento: Factor w/  4 levels "MT500","MT1000",...: 1 1 1 1 1 2 2 2 2 2 ...
 $ Rendimiento  : int  12000 14005 13508 9503 14004 11502 13006 13252 14253 15003 ...
 $ Acelerador    : Factor w/  4 levels "NA-NA","NA-SW",...: 1 1 1 1 1 1 1 1 1 1 ...

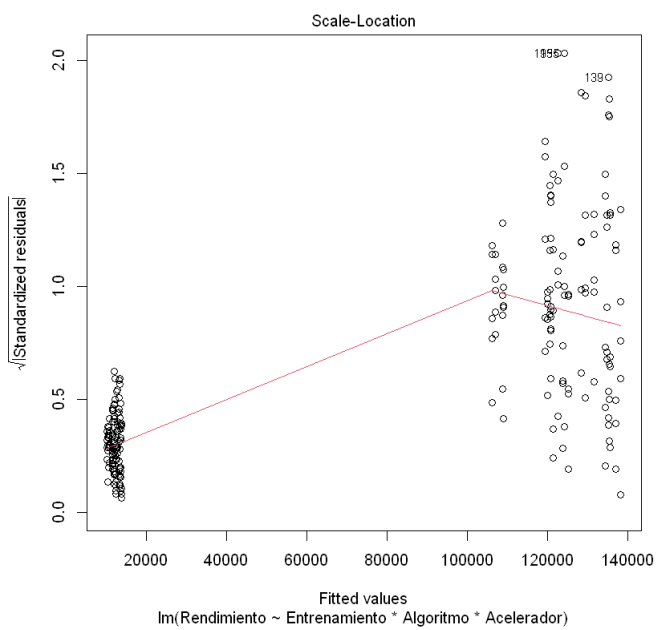
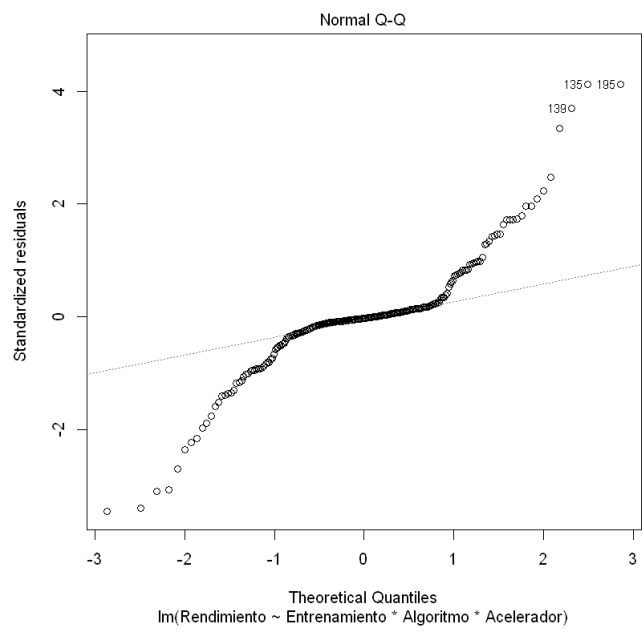
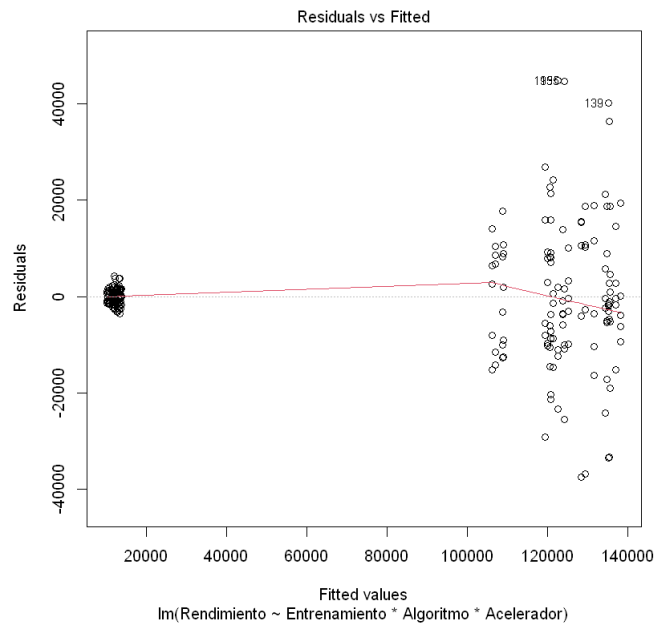
      Algoritmo Entrenamiento Rendimiento Acelerador
Algoritmo A:80  MT500      :60      Min.   : 9000  NA-NA:60
Algoritmo B:80  MT1000     :60     1st Qu.: 12236  NA-SW:60
Algoritmo C:80  MT5000     :60     Median : 53796  HW-NA:60
                  MT50000    :60     Mean   : 68432  HW-SW:60
                  3rd Qu.:124309
                  Max.    :175156
```



A anova: 8 × 4

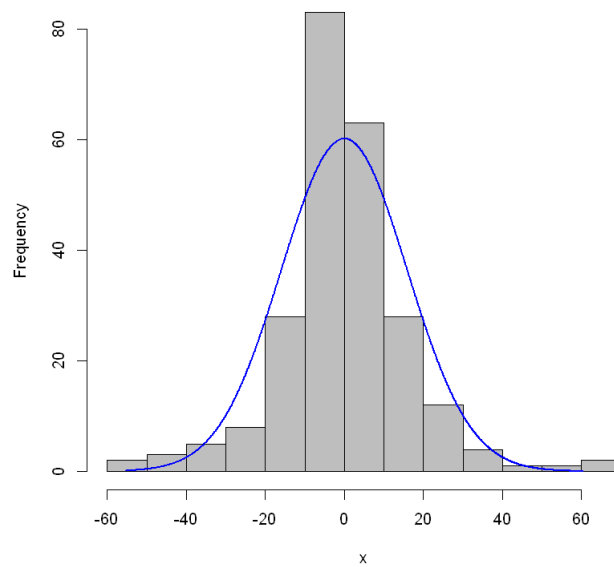
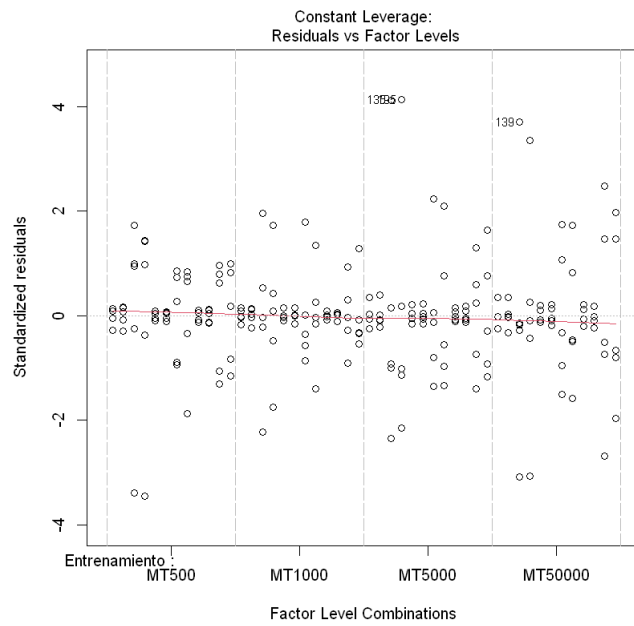
	Sum Sq	Df	F value	Pr(>F)
	<dbl>	<dbl>	<dbl>	<dbl>
Entrenamiento	3105643182	3	7.049149e+00	1.607907e-04
Algoritmo	3384497412	2	1.152313e+01	1.878690e-05
Acelerador	755617765537	3	1.715091e+03	2.580204e-138
Entrenamiento:Algoritmo	248909753	6	2.824861e-01	9.447359e-01
Entrenamiento:Acelerador	2002310059	9	1.514939e+00	1.449261e-01
Algoritmo:Acelerador	2278154237	6	2.585462e+00	1.970571e-02
Entrenamiento:Algoritmo:Acelerador	202043559	18	7.643264e-02	9.999999e-01
Residuals	28196478254	192	NA	NA

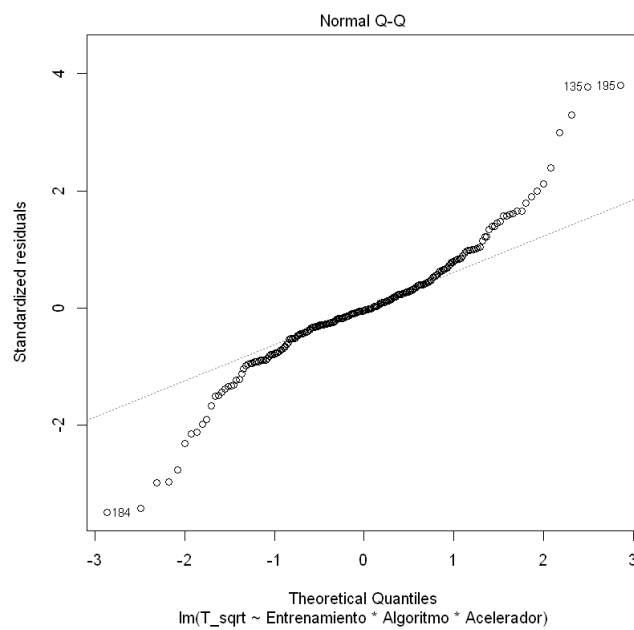
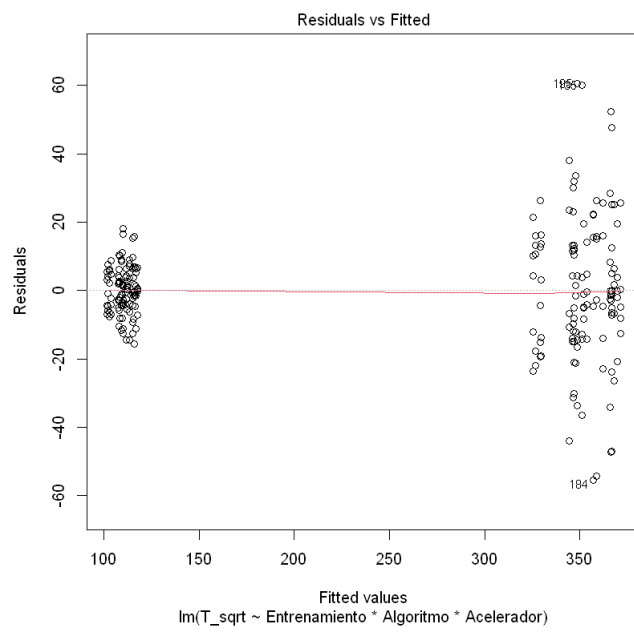
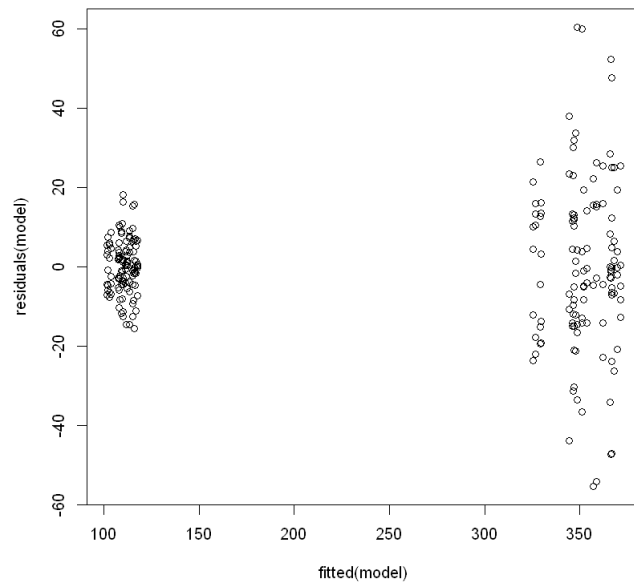


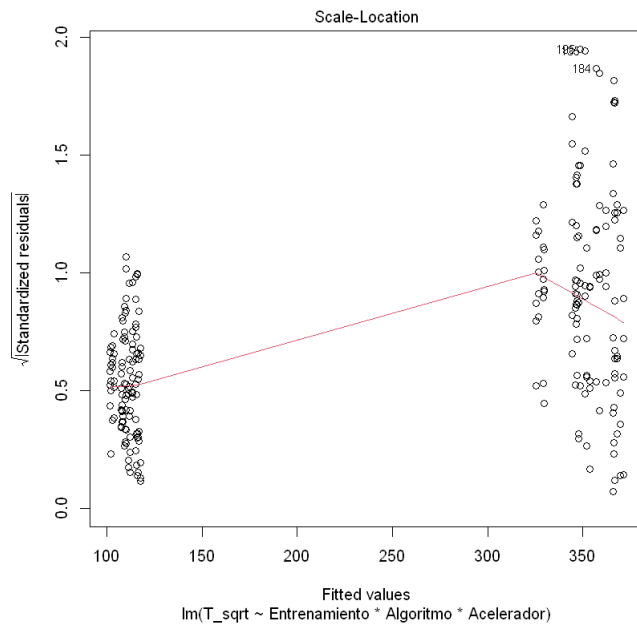


A anova: 8 × 4

	Sum Sq	Df	F value	Pr(>F)
	<dbl>	<dbl>	<dbl>	<dbl>
Entrenamiento	9545.0129	3	1.008357e+01	3.355619e-06
Algoritmo	9671.1896	2	1.532531e+01	6.683816e-07
Acelerador	3492852.0776	3	3.689930e+03	1.923920e-169
Entrenamiento:Algoritmo	767.1671	6	4.052266e-01	8.749951e-01
Entrenamiento:Acelerador	2222.5828	9	7.826628e-01	6.326460e-01
Algoritmo:Acelerador	2626.0124	6	1.387090e+00	2.216297e-01
Entrenamiento:Algoritmo:Acelerador	254.5840	18	4.482474e-02	1.000000e+00
Residuals	60581.7752	192	NA	NA

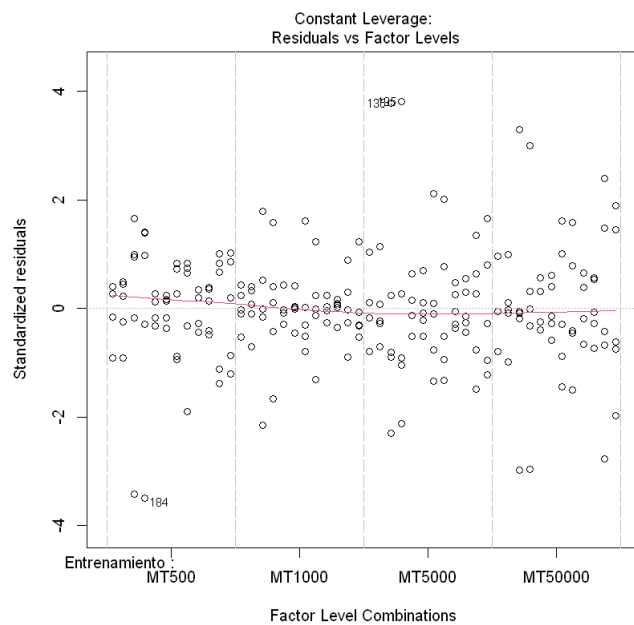


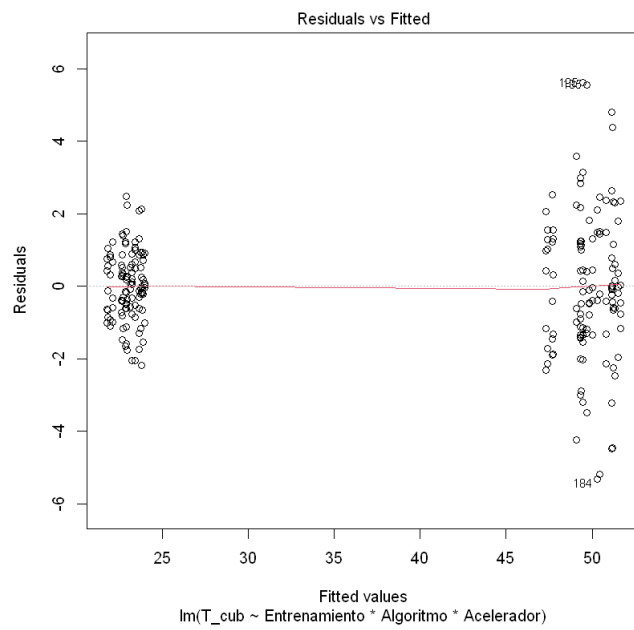
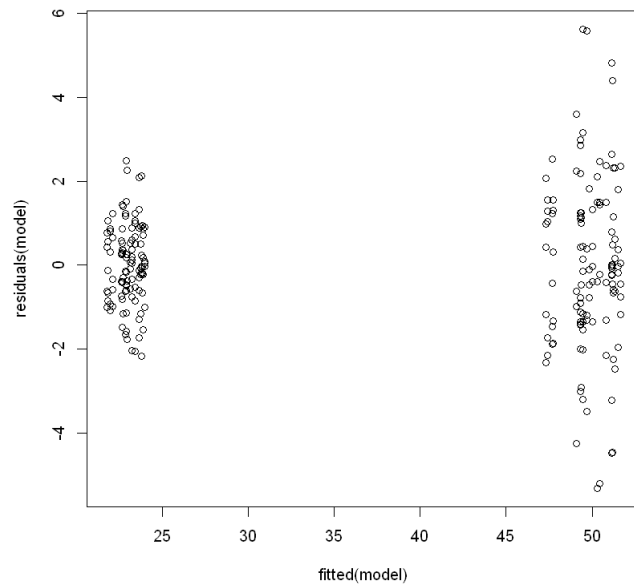
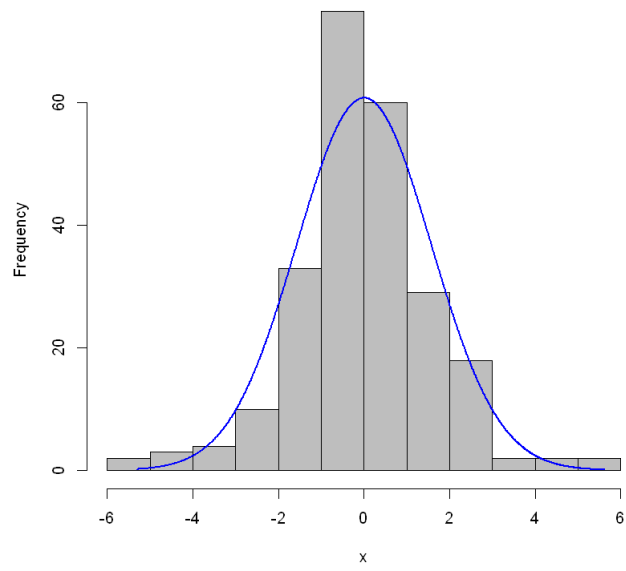


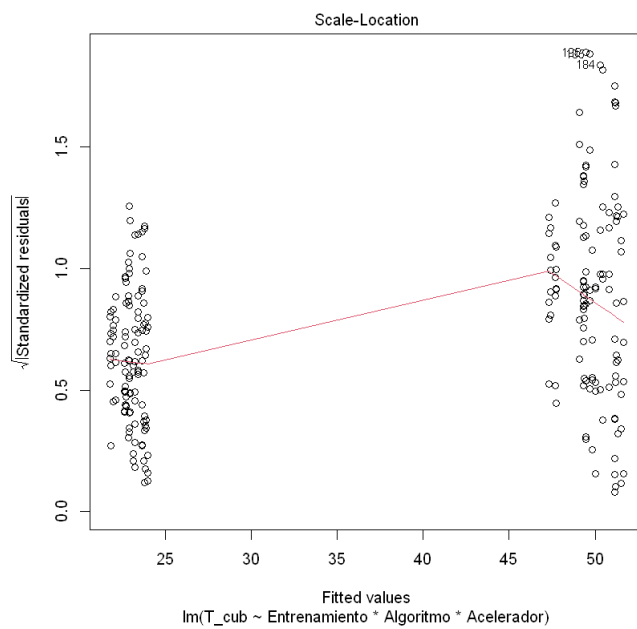
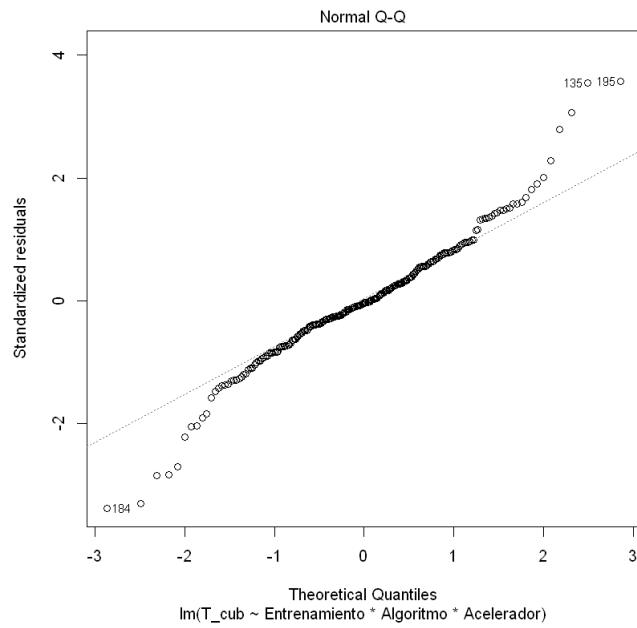


A anova: 8 × 4

	Sum Sq	Df	F value	Pr(>F)
	<dbl>	<dbl>	<dbl>	<dbl>
Entrenamiento	108.368244	3	1.171012e+01	4.428384e-07
Algoritmo	106.700493	2	1.729485e+01	1.241257e-07
Acelerador	43031.289574	3	4.649899e+03	6.181819e-179
Entrenamiento:Algoritmo	8.821202	6	4.766032e-01	8.252271e-01
Entrenamiento:Acelerador	11.320782	9	4.077691e-01	9.299595e-01
Algoritmo:Acelerador	14.439634	6	7.801630e-01	5.864176e-01
Entrenamiento:Algoritmo:Acelerador	1.681881	18	3.029027e-02	1.000000e+00
Residuals	592.271492	192	NA	NA

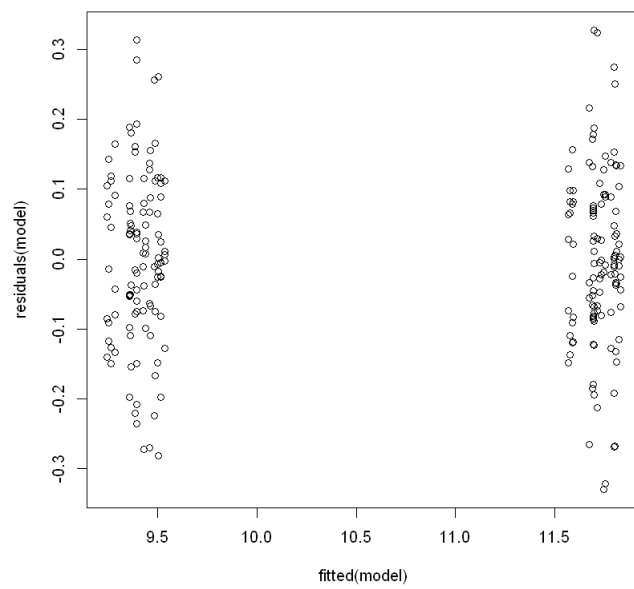
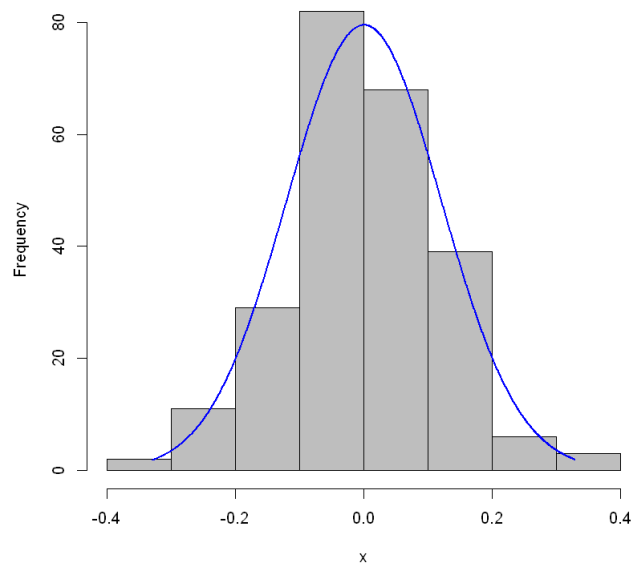
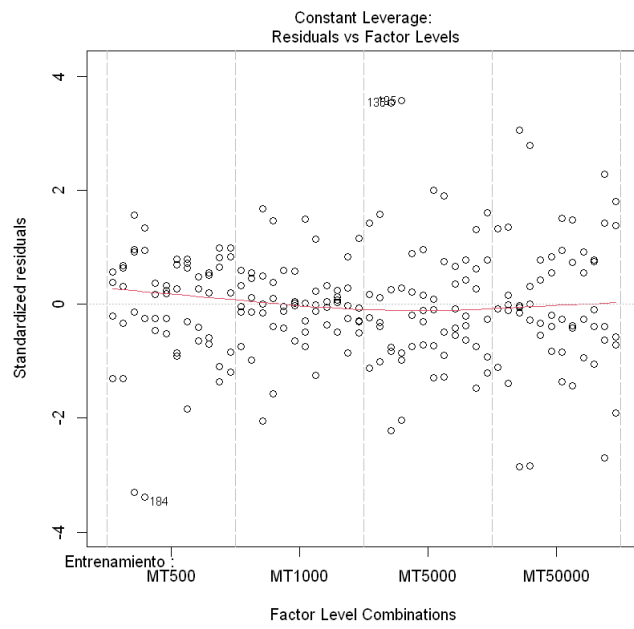


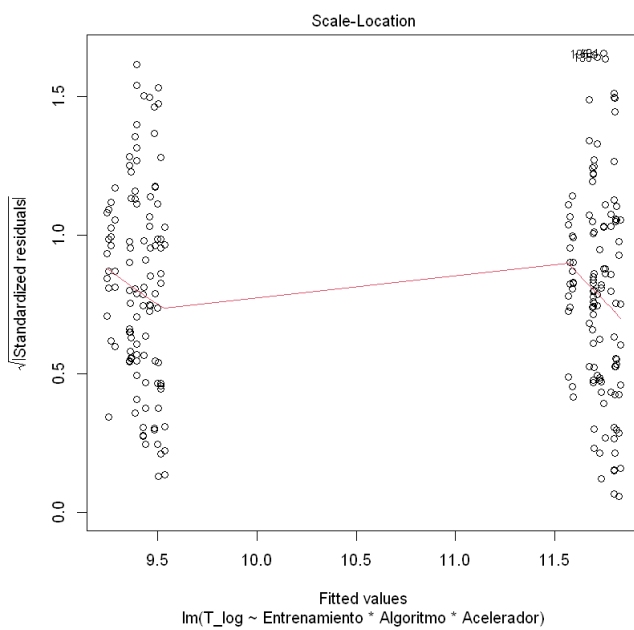
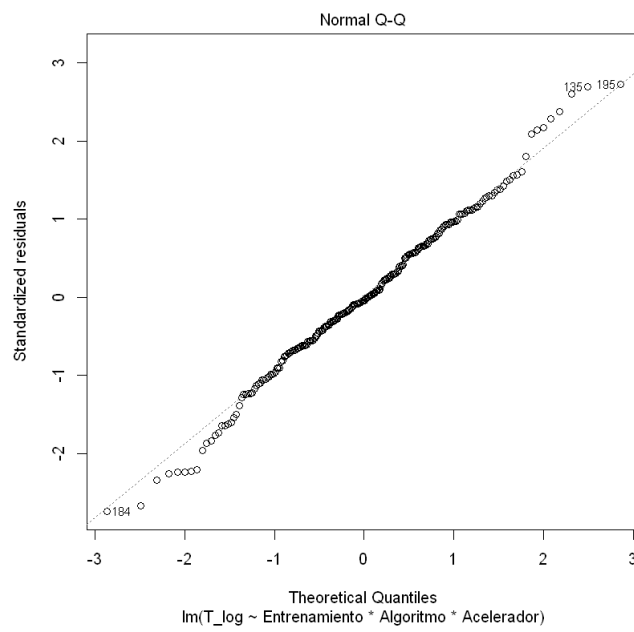
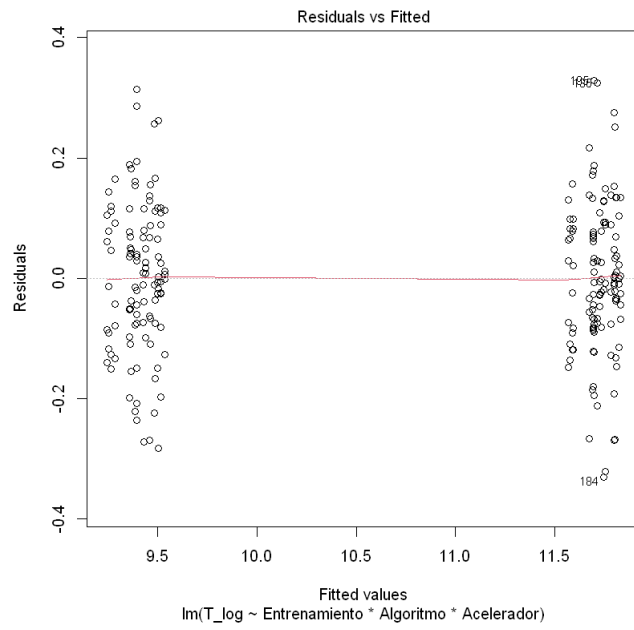




A anova: 8 × 4

	Sum Sq	Df	F value	Pr(>F)
	<dbl>	<dbl>	<dbl>	<dbl>
Entrenamiento	7.831654e-01	3	1.445993e+01	1.565698e-08
Algoritmo	7.260307e-01	2	2.010754e+01	1.178790e-08
Acelerador	3.211197e+02	3	5.928976e+03	6.061320e-189
Entrenamiento:Algoritmo	6.655105e-02	6	6.143810e-01	7.186522e-01
Entrenamiento:Acelerador	5.762868e-03	9	3.546747e-02	9.999953e-01
Algoritmo:Acelerador	3.156218e-03	6	2.913733e-02	9.998927e-01
Entrenamiento:Algoritmo:Acelerador	7.031480e-03	18	2.163756e-02	1.000000e+00
Residuals	3.466308e+00	192	NA	NA





A anova: 2 × 3

	Df	F value	Pr(>F)
	<int>	<dbl>	<dbl>
group	47	0.4057217	0.9997739
	192	NA	NA

NOTE: Results may be misleading due to involvement in interactions

Loading required package: mvtnorm

Loading required package: survival

Loading required package: TH.data

Loading required package: MASS

Attaching package: 'TH.data'

The following object is masked from 'package:MASS':

geyser

Note: adjust = "tukey" was changed to "sidak"
because "tukey" is only appropriate for one set of pairwise comparisons

A summary_emm: 3 × 7

	Algoritmo	lsmean	SE	df	lower.CL	upper.CL	.group
	<fct>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<chr>
3	Algoritmo C	10.48823	0.01502235	192	10.45205	10.52442	a
2	Algoritmo B	10.59413	0.01502235	192	10.55794	10.63031	b
1	Algoritmo A	10.61331	0.01502235	192	10.57713	10.64950	b

NOTE: Results may be misleading due to involvement in interactions

A summary_emm: 4 × 7

	Entrenamiento	lsmean	SE	df	lower.CL	upper.CL	.group
	<fct>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<chr>
3	MT5000	10.50388	0.01734632	192	10.46966	10.53809	a
1	MT500	10.51543	0.01734632	192	10.48122	10.54964	a
4	MT50000	10.60299	0.01734632	192	10.56878	10.63721	b
2	MT1000	10.63860	0.01734632	192	10.60438	10.67281	b

NOTE: Results may be misleading due to involvement in interactions

A summary_emm: 4 × 7

	Acelerador	lsmean	SE	df	lower.CL	upper.CL	.group
	<fct>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<chr>
1	NA-NA	9.398931	0.01734632	192	9.364718	9.433145	a
2	NA-SW	9.418122	0.01734632	192	9.383908	9.452336	a
3	HW-NA	11.719681	0.01734632	192	11.685468	11.753895	b
4	HW-SW	11.724163	0.01734632	192	11.689950	11.758377	b

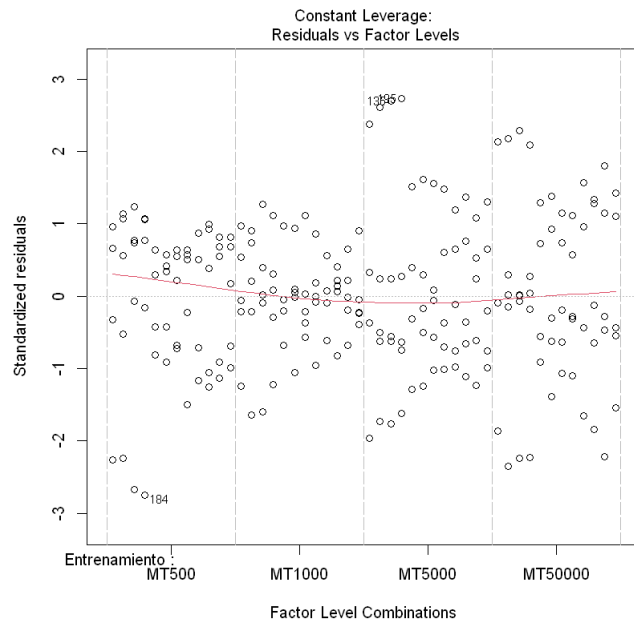
A data.frame: 12 × 11

Entrenamiento	Algoritmo	n	mean	sd	min	Q1	median	Q3	max	se
<fct>	<fct>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>
MT500	Algoritmo A	20	10.599	1.195	9.159	9.523	10.508	11.843	11.905	0.267
MT1000	Algoritmo A	20	10.657	1.184	9.319	9.529	10.617	11.809	11.954	0.265
MT5000	Algoritmo A	20	10.550	1.200	9.159	9.351	10.603	11.641	12.037	0.268
MT50000	Algoritmo A	20	10.647	1.196	9.220	9.482	10.647	11.800	12.073	0.267
MT500	Algoritmo B	20	10.527	1.201	9.251	9.393	10.476	11.729	11.773	0.269
MT1000	Algoritmo B	20	10.677	1.185	9.406	9.526	10.678	11.818	11.968	0.265
MT5000	Algoritmo B	20	10.538	1.195	9.211	9.363	10.579	11.661	11.888	0.267
MT50000	Algoritmo B	20	10.634	1.195	9.321	9.442	10.654	11.771	11.941	0.267
MT500	Algoritmo C	20	10.420	1.200	9.105	9.306	10.411	11.622	11.693	0.268
MT1000	Algoritmo C	20	10.581	1.180	9.339	9.443	10.574	11.715	11.834	0.264
MT5000	Algoritmo C	20	10.423	1.191	9.133	9.240	10.434	11.574	11.748	0.266
MT50000	Algoritmo C	20	10.528	1.195	9.160	9.355	10.478	11.640	11.892	0.267

Warning message:

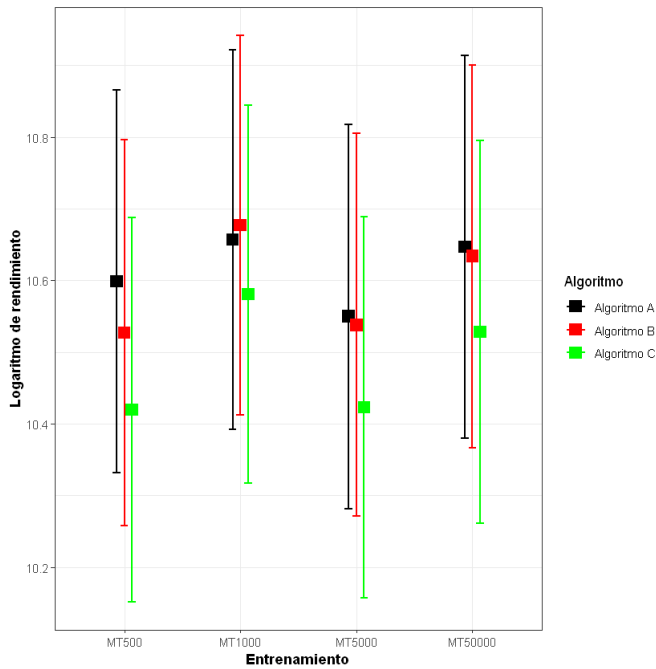
"Using `size` aesthetic for lines was deprecated in ggplot2 3.4.0.

! Please use `linewidth` instead."



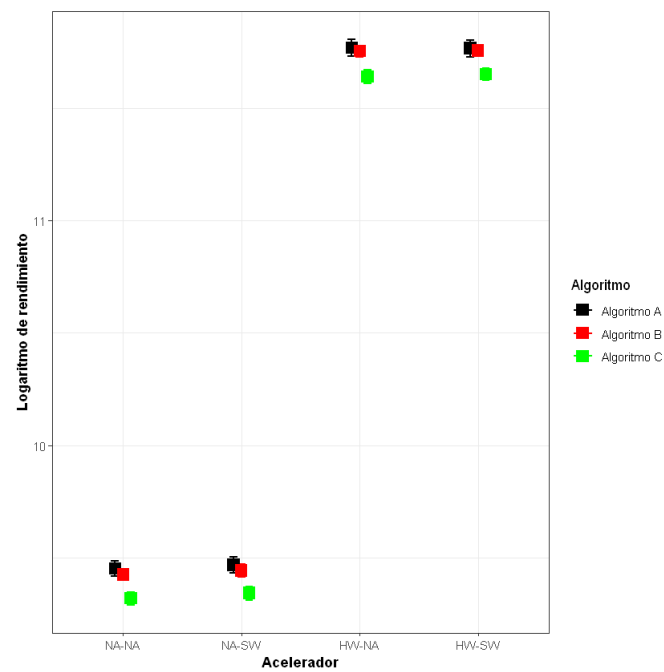
A data.frame: 12 × 11

Acelerador	Algoritmo	n	mean	sd	min	Q1	median	Q3	max	se
<fct>	<fct>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>
NA-NA	Algoritmo A	20	9.452	0.153	9.159	9.351	9.473	9.547	9.741	0.0342
NA-SW	Algoritmo A	20	9.468	0.166	9.184	9.329	9.497	9.590	9.764	0.0371
HW-NA	Algoritmo A	20	11.768	0.169	11.436	11.644	11.795	11.850	12.073	0.0378
HW-SW	Algoritmo A	20	11.766	0.167	11.419	11.655	11.794	11.856	12.053	0.0373
NA-NA	Algoritmo B	20	9.425	0.113	9.211	9.345	9.403	9.511	9.633	0.0253
NA-SW	Algoritmo B	20	9.443	0.123	9.244	9.363	9.421	9.540	9.653	0.0275
HW-NA	Algoritmo B	20	11.753	0.109	11.577	11.684	11.759	11.816	11.968	0.0244
HW-SW	Algoritmo B	20	11.756	0.115	11.517	11.673	11.769	11.830	11.941	0.0257
NA-NA	Algoritmo C	20	9.320	0.128	9.105	9.217	9.317	9.419	9.547	0.0286
NA-SW	Algoritmo C	20	9.343	0.136	9.114	9.232	9.373	9.451	9.547	0.0304
HW-NA	Algoritmo C	20	11.639	0.137	11.410	11.571	11.649	11.719	11.892	0.0306
HW-SW	Algoritmo C	20	11.651	0.118	11.472	11.553	11.674	11.703	11.863	0.0264



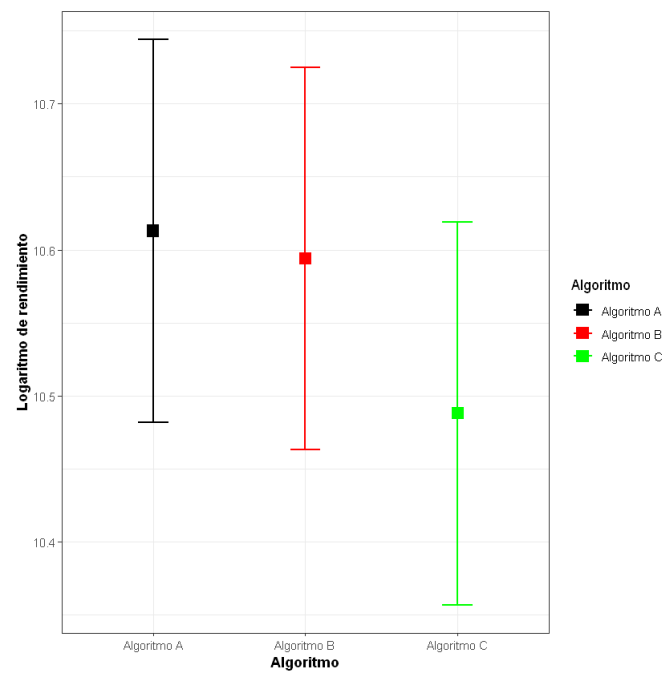
A data.frame: 3 × 10

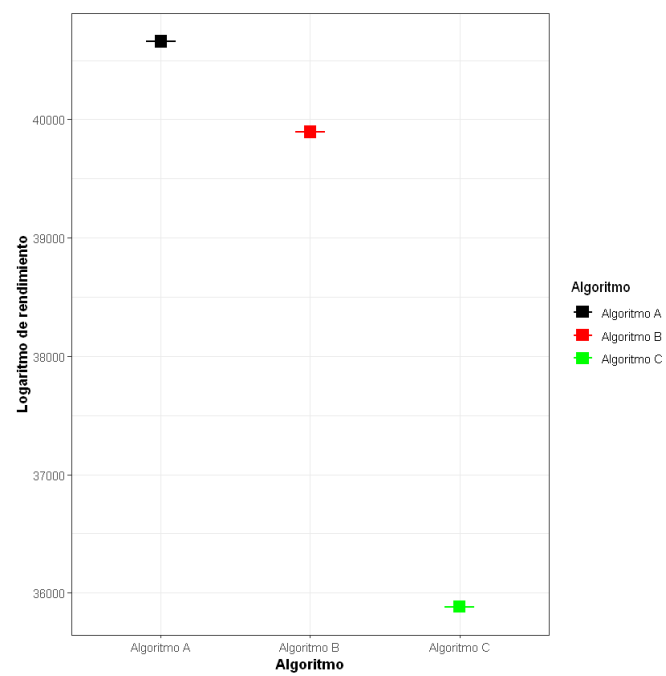
Algoritmo	n	mean	sd	min	Q1	median	Q3	max	se
<fct>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>
Algoritmo A	80	10.613	1.172	9.159	9.482	10.591	11.792	12.073	0.131
Algoritmo B	80	10.594	1.173	9.211	9.413	10.585	11.766	11.968	0.131
Algoritmo C	80	10.488	1.171	9.105	9.348	10.478	11.656	11.892	0.131



A data.frame: 3 × 10

Algoritmo	n	mean	sd	min	Q1	median	Q3	max	se
<fct>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>
Algoritmo A	80	40660.00	3.228443	9.159	9.482	10.591	11.792	12.073	0.361
Algoritmo B	80	39894.75	3.231673	9.211	9.413	10.585	11.766	11.968	0.361
Algoritmo C	80	35882.32	3.225216	9.105	9.348	10.478	11.656	11.892	0.361





In []: